

Curso de Python

1. Generalidades

Python es un lenguaje de **programación de alto nivel y potencial** que se utiliza para desarrollar aplicaciones de todo tipo. A diferencia de otros lenguajes como C++ o Java o .NET, se trata de un **lenguaje interpretado**, es decir, que no es necesario compilarlo para ejecutar las aplicaciones escritas en Python.

Python es un lenguaje sencillo de leer y escribir debido a su **alta similitud con el lenguaje humano en Inglés**. Además, se trata de un lenguaje **multiplataforma de código abierto** y, por lo tanto, gratuito, lo que permite desarrollar software sin límites.

¿De Qué te gustaría trabajar?

Una gran cantidad de tecnologías se llevan muy bien con Python debido a su sencillez y a su gran potencia para el tratamiento de datos, algo que sin duda ha hecho resurgir este lenguaje a nivel laboral, donde cada vez son más las empresas que solicitan expertos en Python.

- **Juegos y gráficos 3D**: Python también posee una gran capacidad para manejar gráficos 3D gracias la gran cantidad de marcos de trabajo y herramientas existentes. PyGame, Blender , Arcade o Battlefield 2.
- **Data analytics y big data**: El uso de Python está muy extendido en dos áreas que han estado, y estarán, en boca de todos: el análisis de datos y el big data. Su simplicidad y su gran número de bibliotecas de procesamiento de datos hacen que Python sea ideal a la hora de analizar y gestionar una gran cantidad de datos en tiempo real.
- **Data mining**: La minería de datos o data mining es un proceso que permite analizar grandes bases de datos con el objetivo de predecir futuras tendencias. Se trata de un proceso complejo al que Python puede arrojar luz a través de la limpieza y organización de datos y del uso de algoritmos de aprendizaje automático que simplifica el análisis de datos.
- **Data science**: Tras la creación de los motores numéricos como “Pandas” o “NumPy”, Python está desbancando MATLAB, un lenguaje utilizado por científicos a la hora de trabajar con un gran número de datos.
- **Inteligencia artificial**: Seguro que durante los últimos años has oído hablar muchísimo de la inteligencia artificial (IA). Gran parte de su avance se debe a Python. Su facilidad de escritura y su robustez han convertido a Python en el aliado perfecto de la IA.
- **Blockchain**: La base de datos distribuida Blockchain, conocida mundialmente por ser la base sobre la que se sustentan las criptomonedas, también funciona muy bien junto a Python.
- **Machine learning**: El machine learning o aprendizaje automático es otra de las tecnologías que está cambiando el mundo tal y como lo conocemos. La robótica y la IA son ahora capaces de aprender por sí mismas a medida que van procesando más y más datos.
- **Desarrollo web**: Python también permite desarrollar webs complejas en menos líneas de código, lo que permite que estas sean más ligeras y optimizadas. Django es uno de los frameworks de Python más populares de la actualidad.

Para poder utilizar Python necesitamos instalar Python que lo descargamos desde <https://www.python.org/> y contar con un Entorno de Desarrollo Integrado (IDE), nosotros utilizaremos pycharm versión community que es sin costo y lo podemos buscar en Google e instalar.

Algunas características son:

- Los archivos de Python tienen extensión “**py**”.
- Las instrucciones **no** tienen indicación de final.
- Es fundamental que el código escrito este correctamente **indentado**.
- Las instrucciones se pueden escribir en cadena y se ejecutan de derecha a izquierda.

2. Bloque de sentencias

Para indicar un bloque de código en Python, debes **indentar** cada línea del bloque en la misma cantidad. Los dos bloques de código en nuestro ejemplo sentencia if están indentados con cuatro espacios que es una cantidad típica de sangría para Python.




Indentación

- **Aumentar la indentación** sirve para indentar luego de un enunciado **if** o **for** (después:)
- **Mantener la indentación** sirve para indicar el **alcance** del bloque (qué líneas son afectadas por **if/for**)
- **Reducir la indentación** permite regresarla al nivel del enunciado **if** o **for** para indicar el final del bloque
- Las **líneas en blanco** son ignoradas y no afectan la **indentación**
- Los **comentarios** en una línea en sí mismos se ignoran en lo que respecta a la **indentación**

3. Comentarios

Para que el programa tome una línea como un comentario debemos utilizar el signo **#**. Y si queremos comentar varias líneas debemos usar las **'''** comillas simples 3 veces como se muestra en el ejemplo de abajo.



```
C:\code\python\CursoPython\1-primeros-pasos\primeros_pasos.py - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help

1 #Comentario sencillo
2
3 '''Comentario
4 entre
5 líneas'''
6
7 print("-----")
8 print("Iniciamos nuestro curso de Python")
9 print("-----")
10 print("En código electrónica")
11
12 #Ecuación matemática
13 print(2 + 3)
```

4. Ingreso de datos

Para que el programa nos permita ingresar un dato por la pantalla se utiliza la sentencia `input`.

Input ("Ingrese su Nombre")

5. Errores de Ortografía

Pycharm tiene un corrector de ortografía en inglés que debemos desactivar ya que vamos a ingresar muchas palabras en castellano que el IDE tomara como erróneas y para que esto no suceda debemos ir al menú `File`, luego `setting`, después `editor`, opción `Inspections`, en `ProofReading`, opción `type` y desactivarla.

6. Paquetes Adicionales de Python

Existen módulos y paquetes adicionales a Python para realizar algunos programas específicos como son los juegos o acceso a las bases de datos que son archivos `.py`, que almacenan funciones, variables y clases, y pueden ser importados por otros archivos.

Los paquetes agrupan estos módulos en carpetas, de los cuales uno debe ser `__init__.py`

En Python también existe un instalador de paquetes que debemos mantener actualizado y tener al día la última versión del instalador de paquetes de Python debemos ejecutar el siguiente comando **`pip install --upgrade pip`** desde la línea de comando del sistema:

```
C:\Users\zanettig>pip install --upgrade pip
Requirement already satisfied: pip in c:\program files\python310\lib\site-packages (22.3.1)
```

Teniendo instalado el instalador de paquetes podemos instalar varios paquetes como pueden ser:

Paquete para juego

Debemos ejecutar la instrucción **`pip install pygame`** desde la consola de comando.

```
C:\Users\zanettig>pip install pygame
Collecting pygame
  Downloading pygame-2.1.2-cp310-cp310-win_amd64.whl (8.4 MB)
    ----- 8.4/8.4 MB 2.3 MB/s eta 0:00:00
Installing collected packages: pygame
Successfully installed pygame-2.1.2
```

Paquete de Base de datos

Luego podemos instalar un paquete que tiene el conector para poder utilizar base de datos mysql desde Python, mediante el comando **pip install pymysql**

```
C:\Users\zanettig>pip install pymysql
Collecting pymysql
  Downloading PyMySQL-1.0.2-py3-none-any.whl (43 kB)
----- 43.8/43.8 kB 1.1 MB/s eta 0:00:00
Installing collected packages: pymysql
Successfully installed pymysql-1.0.2
```

Después la forma de conexión es la siguiente:

```
import pymysql
miConexion = pymysql.connect(host="localhost", user="root",
                             passwd="admin", database="world")
print(cnn)
```

Datos de conexión

```
host="localhost", # 127.0.0.1, o IP del equipo remoto
user="root", #nombre del usuario de la BD, con los permisos necesarios
passwd="admin", #Contraseña del usuario
database="world" #Nombre de la BD
```